Encoding Relation Requirements for Relation Extraction: A Semantic Loss Perspective

Anonymous EMNLP submission

Abstract

Identifying relation between entities is crucial for knowledge base population, question answering and etc. Most existing relation extractors make predictions purely based on the input natural language text, and ignore the argument type and cardinality constraints required by each relation. In this paper, we propose to format these constraints into propositional logic, and incorporate them to existing relation extractors by introducing a semantic loss. Our method does not introduce extra cost to the prediction phase and is plug-and-play for most existing relation extraction models. Experiments on both English and Chinese datasets show that our method improves the base model by a clear margin.

1 Introduction

Relation extraction (RE) aims at identifying relation between entity pair from raw text, and its success can benefit many knowledge base (KB) related tasks like KB population, question answering (QA) and etc (Suchanek et al., 2013).

In the literature, RE is usually investigated in the distant supervision (DS) paradigm, where datasets are automatically constructed by aligning existing KB *(subject, relation, object)* triples with a large corpus, and considers sentences containing the subject and object entity in a triple as evidence for the corresponding relation (Riedel et al., 2010). (YY:need to revise) To alleviate the sentence level noise in the automatically constructed dataset, RE is often considered in the multi-instance learning (MIL) framework where all the sentences containing the target subject and object are packed into a ***sentence bag***, and relation extractors take in these sentences to predict the relation for this entity pair. Following this framework, Zeng et al. (2015) uses piecewise convolutional neural network (PCNN) to handle the extraction task, Lin et al. (2016) introduces attention mechanism for better noise tolerance, Ye et al. (2017) makes further improvements by learning co-occurrence tendency between relations via learning to rank. (YY:need to revise)

Typically, most existing relation extractors relies only on input sentences to make predictions, and ignores the constraints required by each relation. Take the relation *Capital* for example, it expects its subject to be a country and its object to be a city. And in most cases, it also expects a city to be the capital of one country only. These kinds of constraints can help us identify inconsistent predictions and thereby improve the extraction performance.

However, properly utilizing these clues is non-trivial, since many KBs do not have a well-defined typing system or a cardinality specification for target relations. Chen et al (2014) deals with this challenge by implicitly mining such requirements from data. Specifically, while collecting cardinality requirements directly from data, they evades the tricky argument type constraints by collecting relation pairs that have the same subject or object type instead, which do not require a concrete specification for argument types. (YY:need to revise) Then, they use integer linear programming (ILP) to resolve the predictions that are inconsistent regarding the constraints. However, since ILP operates at the post-processing phase and needs an adequate number of predictions to correct results, their method typically requires more time during prediction, and could not be applied to online applications. Besides, since ILP does not change the prediction scores of the base model, it tends to leave the predictions corrected by the constraints with low scores, which hurts the performance in terms of the precision-recall curve criterion.

To overcome the problems of ILP, we propose to incorporate these constraints by introducing a semantic loss to penalize inconsistent predictions during training. Specifically, we consider the type and cardinality constraints as propositional logic constraints, and use the semantic loss framework (Xu et al., 2017) to convert them into a loss term. Compared to other methods of enforcing logical constraints like the teacher-student network (Hu et al., 2016) that relies on fuzzy relaxation of the constraints, semantic loss possesses the precise meaning of the constraints and is fully differentiable since it directly uses the predicted probability to construct the loss. In this way, the base model is encouraged to find more textual clues when detecting conflicts, which leads to better prediction ability of the model, and the prediction phase incurs no extra costs. Further, since we only add a loss term to the base model, our method is plug-and-play for most relation extraction models.

We conduct experiments on both English and Chinese datasets, and the experimental results show that our method can clearly improve the base model, and delivers comparable performance compared to the ILP method.

2 Our Approach

First, we briefly introduce our base relation extraction model, then we describe our constraints in detail and how to incorporate these constraints during training.

2.1 Base Model

While our method is compatible to most existing relation extractors, in this paper, we take the attentive convolutional neural network (Lin et al., 2016) as our base model, which is currently one of the most widely used extractor in RE. It operates in the MIL framework, specifically it uses convolutional neural network for sentence embedding, and an attention layer to selectively aggregate sentence embeddings into a sentence bag embedding, which is then fed to a softmax classifier to generate the predicted relation distribution, $\boldsymbol{p}'$.

2.2 Relation Constraints

Our relation constraints are similar to those used by Chen et al. (2014). Specially, our relation constraints are defined over each combination of two entity pairs: $(subj_1, r_1 obj_1)$ and $(subj_2, r_2, obj_2)$ (we use $subj$, obj and r to denote subject, object and relation for entity pair, respectively, in the rest of the paper). We derive type constraints and cardinality constraints from existing KB to implicitly capture the expected type and cardinality requirements for the arguments of a relation.

Type Constraint If the subject (or object) set of relation r_1 in KB has a large intersection with those of r_2, then we consider r_1 and r_2 to have the same expected subject (or object) type. We thereby assign relation pairs (r_1, r_2) into $\boldsymbol{C}^{ts}$ if they have the same subject type, $\boldsymbol{C}^{to}$ if they have the same object type, $\boldsymbol{C}^{tso}$ if the subject type of one relation is the same as the object type of the other. The idea is that, if the predicted relations of two triples require the same entity to belong to different types, then at least one of the prediction must be wrong.

Cardinality Constraint Given a subject (or object), some relations should have only one object (or subject). For example, the relation *Capital* would expect only one object for a given subject. Following this observation, we assign relation r into $\boldsymbol{C}^{cs}$ if it can have multiple subjects for a given object, $\boldsymbol{C}^{co}$ if it can have multiple objects for a given subject.

Thus we finally get 5 sub-categories of constraint sets $\boldsymbol{C}^{\boldsymbol{\Phi}}$, where $\boldsymbol{\Phi} = \{ts, to, tso, cs, co\}$

2.3 Incorporate Constraints for Training

In this section, we demonstrate our method of converting the relation constraints into a loss term using the semantic loss framework (Xu et al., 2017). We also introduce a principled variant of our method to speed up the training process with only minor drop in the prediction ability.

Relation Constraint Loss Semantic loss is a general framework that can encode propositional logic constraints as a loss term in a principled way. Concretely, the semantic loss $L^s(\boldsymbol{C}, \boldsymbol{p})$ is defined as:

$$L^s(\boldsymbol{C}, \boldsymbol{p}) = -log \sum_{\boldsymbol{x} \models \boldsymbol{C}} \prod_{i:\boldsymbol{x} \models X_i} p_i \prod_{i:\boldsymbol{x} \models \neg X_i} (1 - p_i) \tag{1}$$

where $\boldsymbol{C}$ is a set of constraints that are defined over variables $\boldsymbol{X} = \{X_1, X_2, ..., X_n\}$, p_i is the predicted probability of X_i which is defined in Eq. 3, $\boldsymbol{x} \models \boldsymbol{C}$ refers to the assignment $\boldsymbol{x} = \{x_1, x_2, ..., x_n\}$ of variables $\boldsymbol{X}$ that satisfies the constraints in $\boldsymbol{C}$, and $i : \boldsymbol{x} \models X_i$ (or $i : \boldsymbol{x} \models \neg X_i$) refers to the indices i where x_i is set to true (or false) in assignment $\boldsymbol{x}$.

As for our type constraints, the assignment $\boldsymbol{x}$ of variables $\boldsymbol{X} \in \{0,1\}^R$, where R is the number of relations, is derived by a pair of relations $(r_1, r_2) \in \boldsymbol{C}^{ts} \cup \boldsymbol{C}^{to} \cup \boldsymbol{C}^{tso}$. Specifically, $x_i \in \boldsymbol{x}$ equals 1 only when r_1 or r_2 is the i^{th} relation, and all $x_i \in \boldsymbol{x}$ are set to 0 when $r_1 = r_2$.

As for our cardinality constraints, the assignment $\boldsymbol{x}$ of variables $\boldsymbol{X} \in \{0,1\}^R$ is derived by one relation $r \in \boldsymbol{C}^{cs} \cup \boldsymbol{C}^{co}$. Specifically, $x_i \in \boldsymbol{x}$ equals 1 only when r is the i^{th} relation.

Training Procedure Here we introduce the learning and optimization details of our model. Our objective function consists of two parts:

$$J(\theta) = J_{en}(\theta) + J_{SL}(\theta) \tag{2}$$

where $J_{en}(\theta)$ is the original cross-entropy classification loss, and $J_{SL}(\theta)$ is our semantic loss.

For any combinations of two entity pairs in a batch, we use the left and right part of Eq. 3 to get the probability vector $\boldsymbol{p}$ for type constraints and cardinality constraints in Eq. 1, respectively.

$$p_i^t = 1 - (1 - p'_{1i})(1 - p'_{2i}); \quad p_i^c = p'_{1i} p'_{2i} \tag{3}$$

where $\boldsymbol{p}'_1$ and $\boldsymbol{p}'_2$ represents the outputs of our base model for the two entity pairs, p_i^t and p_i^c denotes the predicted probability of variable X_i in the type and cardinality constraints, respectively.

And then we compute the semantic loss terms $\lambda^k m^k L^s(\boldsymbol{C}^k, \boldsymbol{p}), (k \in \boldsymbol{\Phi})$ for each sub-category constraints over two entity pairs. Here λ^k denotes the corresponding weight coefficient and m^k is a 0-1 mask which is set to 1 when the two entity pairs satisfies the conditions of $\boldsymbol{C}^k$.

Finally our semantic loss is defined as follows:

$$J_{SL}(\theta) = \sum_{i<j} \sum_{k \in \boldsymbol{\Phi}} \lambda^k m^k L^s(\boldsymbol{C}^k, \boldsymbol{p}_{ij}^k) \tag{4}$$

where $0 \leq i < j < batch\_size$, $\boldsymbol{p}_{ij}^k$ refers to the probability vector generated from Eq. 3.

Note that, for each loss term, all the relation assignment pairs that satisfies the corresponding constraints are included. Therefore, minimizing these semantic loss terms actually increases the likelihood of all the relation predictions that satisfies our relation constraints.

During training, we iterate by randomly selecting a mini-batch from the training set until converge, and adopt the Adam optimizer (Kingma and Ba, 2014) to minimize the objective function.

Simplified Semantic Loss In Eq. (1), for each entity pair combination, we need to calculate the semantic loss for every $\boldsymbol{x} \models \boldsymbol{C}$, which would be time-consuming since there are many relation assignments that are consistent with the constraint set $\boldsymbol{C}$. However, among these assignments, only the gold standard relation assignment is the one that we desire. Therefore, we simplifies Eq. (1) by only including the gold relation assignment and a few randomly sampled consistent assignments as supplements. We find that, with this simplification, we can significantly speed up the training process at the cost of only minor drop in performance.

3 Experiments

3.1 Experiment Settings

Datasets We evaluate our approach on an English dataset and a Chinese dataset, which are proposed by Chen et al. (2014).

The English dataset is generated by mapping the triples in DBpedia (Bizer et al., 2009) to the sentences in the New York Time corpus. It has 51 relations, about 50k entity tuples, 134k sentences for training and 30k entity tuples, 53k sentences for testing.

The Chinese dataset uses a KB constructed by using the Infoboxes of HudongBaiKe, and aligns its triples to a corpus collected from four chinese economic newspapers. It contains 28 relations, about 60k entity tuples, 120k sentences for training and 40k tuples, 83k sentences for testing.

We do not use Riedel's dataset (Riedel et al., 2010), which is commonly used in RE, because Chen et al. (2014) have already proven that relation constraints does not work on that dataset.

Hyperparameters We use grid search to determine the optimal parameters. Our base model use convolution window size 3, sentence embedding size 256, position embedding size 5 and batch size 50. The word embedding size is 50 and 300 for the English and Chinese dataset respectively. The loss weights for type constraints are 0.001 and 0.005 for the English and Chinese dataset respectively, and the weight for cardinality constraints is 0.0005 for both the English and Chinese dataset.

3.2 Experimental Results

Following previous work on RE, we use the precision-recall curve as our evaluation criterion.

We compare our semantic loss method (`SL`) along with its simplified version (`SL-simple`) with the base relation extractor (referred to as `Base`, see Sec. 2.1) and the `ILP` method (Chen et al., 2014), which uses `ILP` to incorporate the relation constraints at the post-processing phase.

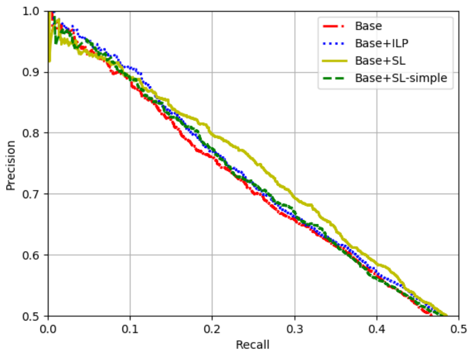

Figure 1: Performance on English Dataset

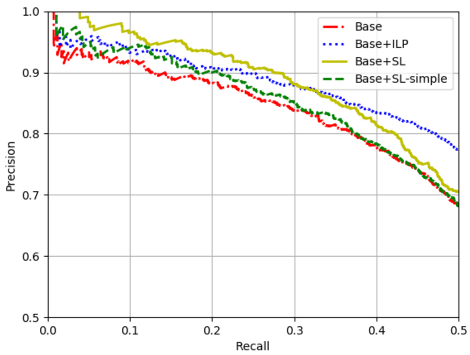

Figure 2: Performance on Chinese Dataset

Compare with Base Model As shown in Fig. 1 and Fig. 2, by encouraging the base relation extractor to make predictions that are consistent to our constraints, our `SL` method clearly improves the baseline relation extractor in both datasets.

Take <*Center For Responsible Lending, North Carolina*> and <*Meredith College, North Carolina*> as an example. `Base` outputs relations *Location* and *CoachedTeam* for these two entity pairs, and our `SL` model predicts *Location* and *State* instead. Note that *Location* requires its subject to be a place, and *CoachedTeam* expects its subject to be a team, so there exists a conflict between the two predictions. `Base` confuses *CoachedTeam* with *State* since many team names are the same as their state names, and these two relations sometimes share similar expressions in the contexts. However, with the help of relation constraints during training, our semantic loss term identifies the conflict and thus encourages the base extractor to focus more on the textual clues about relation *CoachedTeam*.

Compare with ILP Method As shown in Fig. 1, in the English dataset, `SL` performs generally better than `ILP`. We think the superior performance comes from the fact that `SL` functions during training and `ILP` only acts as a post-processor. Therefore, `SL` can encourage the base model to find more textual clues when detecting conflicts, while `ILP` can only find the most probable relation assignment that satisfies the constraints based on the output scores of the base model, which will possibly drop some high-score predictions and thus still leave the correct relation with a low score.

For the Chinese dataset, as shown in Fig. 2, we can see that `SL` performs better in the high precision region.

Note that, in practice, since we often require high-confidence extraction results, the performance of the high-precision region of the PR curve is more important than the low-precision one. Further, recall that different from `ILP`, `SL` does not introduce extra cost during prediction. These experiments indicate that our `SL` method is more effective than `ILP` in practice.

Compare with Simplified Semantic Loss We also show the performance of our simplified semantic loss in Fig. 1 and Fig. 2 (`SL-simple`). We can see that, while inferior to `SL`, `SL-simple` also improves `Base` by a clear margin, and it is also comparable to `ILP` in the English dataset. In our experiments, compared to the original `SL` method, `SL-simple` reduces the extra training time introduced by calculating the semantic loss by 7 times in the English dataset, and 3 times in the Chinese dataset, which has less constraints than those in the English dataset. This indicates that `SL-simple` acts as a good balance of the trade-off between extraction quality and extra training time.

4 Conclusion

In this paper, we introduced a new loss term for `RE` to help resolve the conflicts to type and cardinality constraints among local relation predictions, which is compatible to most existing relation extraction models. Our method does not bring extra cost during prediction, and the experiments show that our method consistently improves the results of the base relation extractor by a clear margin.

References

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165.

Liwei Chen, Yansong Feng, Songfang Huang, Yong Qin, and Dongyan Zhao. 2014. Encoding relation requirements for relation extraction via joint inference. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 818–827.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. *arXiv preprint arXiv:1603.06318*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*, volume 1, pages 2124–2133.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Fabian Suchanek, James Fan, Raphael Hoffmann, Sebastian Riedel, and Partha Pratim Talukdar. 2013. Advances in automated knowledge base construction. *SIGMOD Records journal, March*.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. 2017. A semantic loss function for deep learning with symbolic knowledge. *arXiv preprint arXiv:1711.11157*.

Hai Ye, Wenhan Chao, Zhunchen Luo, and Zhoujun Li. 2017. Jointly extracting relations with class ties via effective deep ranking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1810–1820. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*, pages 1753–1762.